Numerical Methods II

Prof. Mike Giles

mike.giles@maths.ox.ac.uk

Oxford University Mathematical Institute

Quasi-Monte Carlo

As in lecture 6, quasi-Monte Carlo methods offer much greater accuracy for the same computational costs.

Same ingredients:

- Sobol or lattice rule quasi-uniform generators
- PCA to best use QMC inputs for multi-dimensional applications
- randomised QMC to regain confidence interval

New ingredient:

- how best to use QMC inputs to generate Brownian increments

Quasi-Monte Carlo

In lecture 12, expressed expectation as a multi-dimensional integral with respect to unit Normal inputs

$$V = \mathbb{E}[\widehat{f}(\widehat{S})] = \int \widehat{f}(\widehat{S})\ \phi(Z)\ \mathrm{d}Z$$

where $\phi(Z)$ is multi-dimensional unit Normal p.d.f.

Putting $Z_n = \Phi^{-1}(U_n)$ turns this into an integral over a M-dimensional hypercube

$$V = \mathbb{E}[\widehat{f}(\widehat{S})] = \int \widehat{f}(\widehat{S})\ \mathrm{d}U$$

Quasi-Monte Carlo

This is then approximated as

$$N^{-1} \sum_n \widehat{f}(\widehat{S}^{(n)})$$

and each path calculation involves the computations

$$U \rightarrow Z \rightarrow \Delta W \rightarrow \widehat{S} \rightarrow \widehat{f}$$

The key step here is the second, how best to convert the vector Z into the vector ΔW. With standard Monte Carlo, as long as ΔW has the correct distribution, how it is generated is irrelevant, but with QMC it does matter.

Quasi-Monte Carlo

For a scalar Brownian motion $W(t)$ with $W(0)=0$, defining $W_n=W(nh)$, each W_n is Normally distributed and for $j \geq k$

$$\mathbb{E}[W_j\, W_k] = \mathbb{E}[W_k^2] + \mathbb{E}[(W_j - W_k)\, W_k] = t_k$$

since $W_j - W_k$ is independent of W_k.

Hence, the covariance matrix for W is Ω with elements

$$\Omega_{j,k} = \min(t_j, t_k)$$

Quasi-Monte Carlo

The task now is to find a matrix L such that

$$L\,L^T = \Omega = h \begin{pmatrix} 1 & 1 & \dots & 1 & 1 \\ 1 & 2 & \dots & 2 & 2 \\ \dots & \dots & \dots & \dots & \dots \\ 1 & 2 & \dots & M-1 & M-1 \\ 1 & 2 & \dots & M-1 & M \end{pmatrix}$$

We will consider 3 possibilities:

- Cholesky factorisation
- PCA
- Brownian Bridge treatment

Cholesky factorisation

The Cholesky factorisation gives

$$L = \sqrt{h} \begin{pmatrix} 1 & 0 & \dots & 0 & 0 \\ 1 & 1 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 1 & 1 & \dots & 1 & 0 \\ 1 & 1 & \dots & 1 & 1 \end{pmatrix}$$

and hence

$$W_n = \sum_{m=1}^{n} \sqrt{h}\, Z_m \quad \Longrightarrow \quad \Delta W_n = W_n - W_{n-1} = \sqrt{h}\, Z_n$$

i.e. standard MC approach

PCA construction

The PCA construction uses

$$L = U\,\Lambda^{1/2} = \left(\; U_1 \,\middle|\, U_2 \,\middle|\, \ldots \; \right) \begin{pmatrix} \lambda_1^{1/2} & & \\ & \lambda_2^{1/2} & \\ & & \ldots \end{pmatrix}$$

with the eigenvalues λ_n and eigenvectors U_n arranged in descending order, from largest to smallest.

Numerical computation of the eigenvalues and eigenvectors is costly for large numbers of timesteps, so instead use theory due to Åkesson and Lehoczky (1998)

PCA construction

It is easily verified that

$$\Omega^{-1} = h^{-1} \begin{pmatrix} 2 & -1 & & & & & \\ -1 & 2 & -1 & & & & \\ & -1 & 2 & -1 & & & \\ & & \cdots & \cdots & \cdots & & \\ & & & -1 & 2 & -1 & \\ & & & & -1 & 2 & -1 \\ & & & & & -1 & 1 \end{pmatrix}.$$

This looks like the finite difference operator approximating a second derivative, and so the eigenvectors are Fourier modes.

PCA construction

The eigenvectors of both Ω^{-1} and Ω are

$$(U_m)_n = \frac{2}{\sqrt{2M+1}} \sin\left(\frac{(2m-1)\, n\pi}{2M+1}\right)$$

and the eigenvalues of Ω are the reciprocal of those of Ω^{-1},

$$\lambda_m = \frac{h}{4}\left(\sin\left(\frac{(2m-1)\,\pi}{2\,(2M+1)}\right)\right)^{-2}$$

Because the eigenvectors are Fourier modes, an efficient FFT transform can be used (Scheicher, 2006) to compute

$$L\,Z \;=\; U\,\left(\Lambda^{1/2}\,Z\right) \;=\sum_m (\sqrt{\lambda_m}\;Z_m)\,U_m$$

Brownian Bridge construction

The Brownian Bridge construction uses the theory from lecture 10.

The final Brownian value is constructed using Z_1:

$$W_M = \sqrt{T}\, Z_1$$

Conditional on this, the midpoint value $W_{M/2}$ is Normally distributed with mean $\frac{1}{2}W_M$ and variance $T/4$, and so can be constructed as

$$W_{M/2} = \tfrac{1}{2}W_M + \sqrt{T/4}\, Z_2$$

Brownian Bridge construction

The quarter and three-quarters points can then be constructed as

$$\begin{aligned} W_{M/4} &= \tfrac{1}{2}W_{M/2} + \sqrt{T/8}\, Z_3 \\ W_{3M/4} &= \tfrac{1}{2}(W_{M/2} + W_M) + \sqrt{T/8}\, Z_4 \end{aligned}$$

and the procedure continued recursively until all Brownian values are defined.

(This assumes M is a power of 2 – if not, the implementation is slightly more complex)

I have a slight preference for this method because it is particularly effective for European options for which $S(T)$ is very strongly dependent on $W(T)$.

Multi-dimensional Brownian motion

The preceding discussion concerns the construction of a single, scalar Brownian motion.

Suppose now that we have to generate a P-dimensional Brownian motion with correlation matrix Σ between the different components.

What do we do?

Multi-dimensional Brownian motion

First, using either PCA or BB to construct P uncorrelated Brownian paths using

- $Z_1, Z_{1+P}, Z_{1+2P}, Z_{1+3P}, \ldots$ for first path
- $Z_2, Z_{2+P}, Z_{2+2P}, Z_{2+3P}, \ldots$ for second path
- $Z_3, Z_{3+P}, Z_{3+2P}, Z_{3+3P}, \ldots$ for third path
- etc.

This uses the "best" dimensions of Z for the overall behaviour of all of the paths.

Multi-dimensional Brownian motion

Second, define

$$W_n^{corr} = L_\Sigma \, W_n^{uncorr} \quad \Longrightarrow \quad \Delta W_n^{corr} = L_\Sigma \, \Delta W_n^{uncorr}$$

where W_n^{uncorr} is the uncorrelated sequence,
W_n^{corr} is the correlated sequence, and

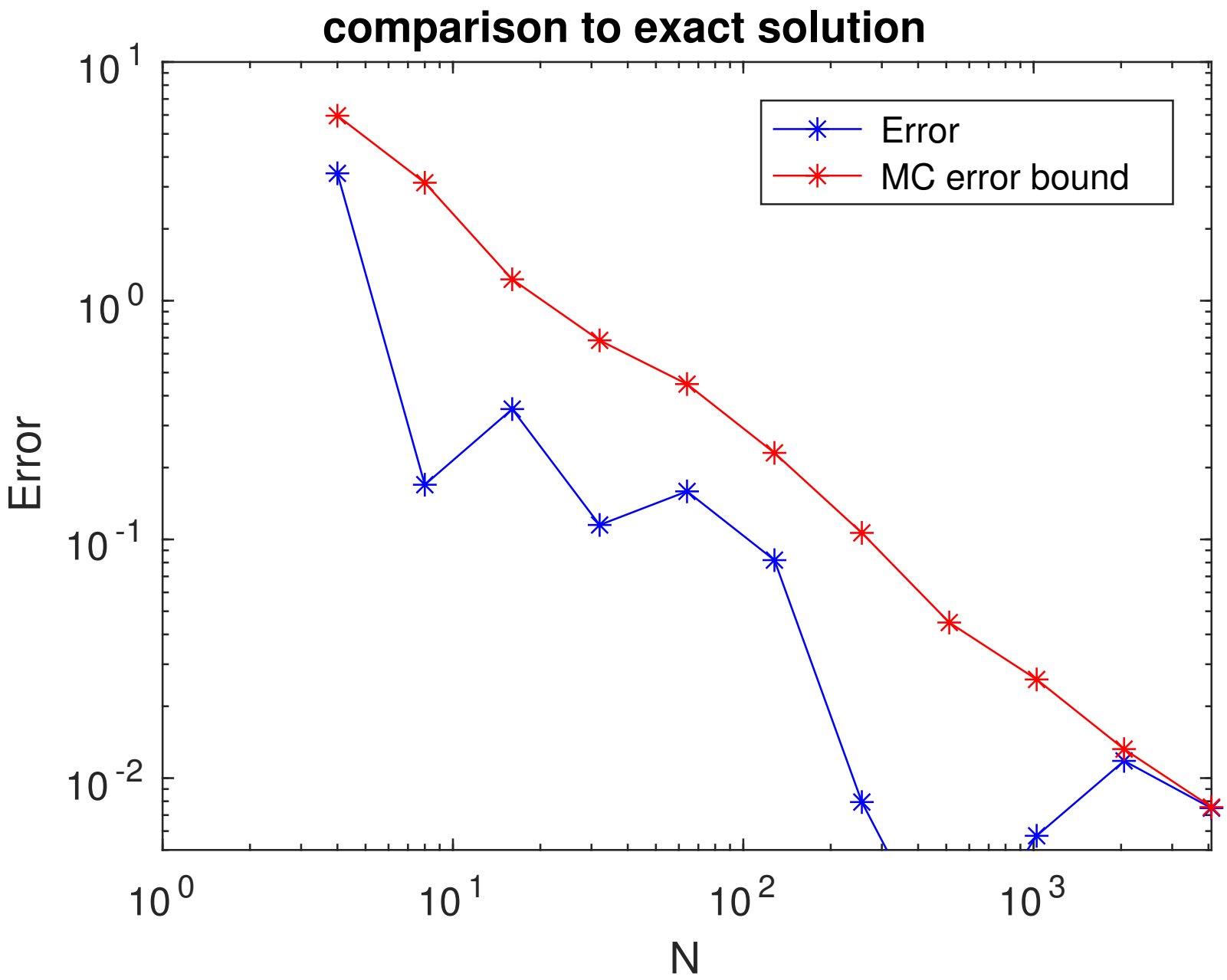
$$L_\Sigma \, L_\Sigma^T = \Sigma$$

Numerical results

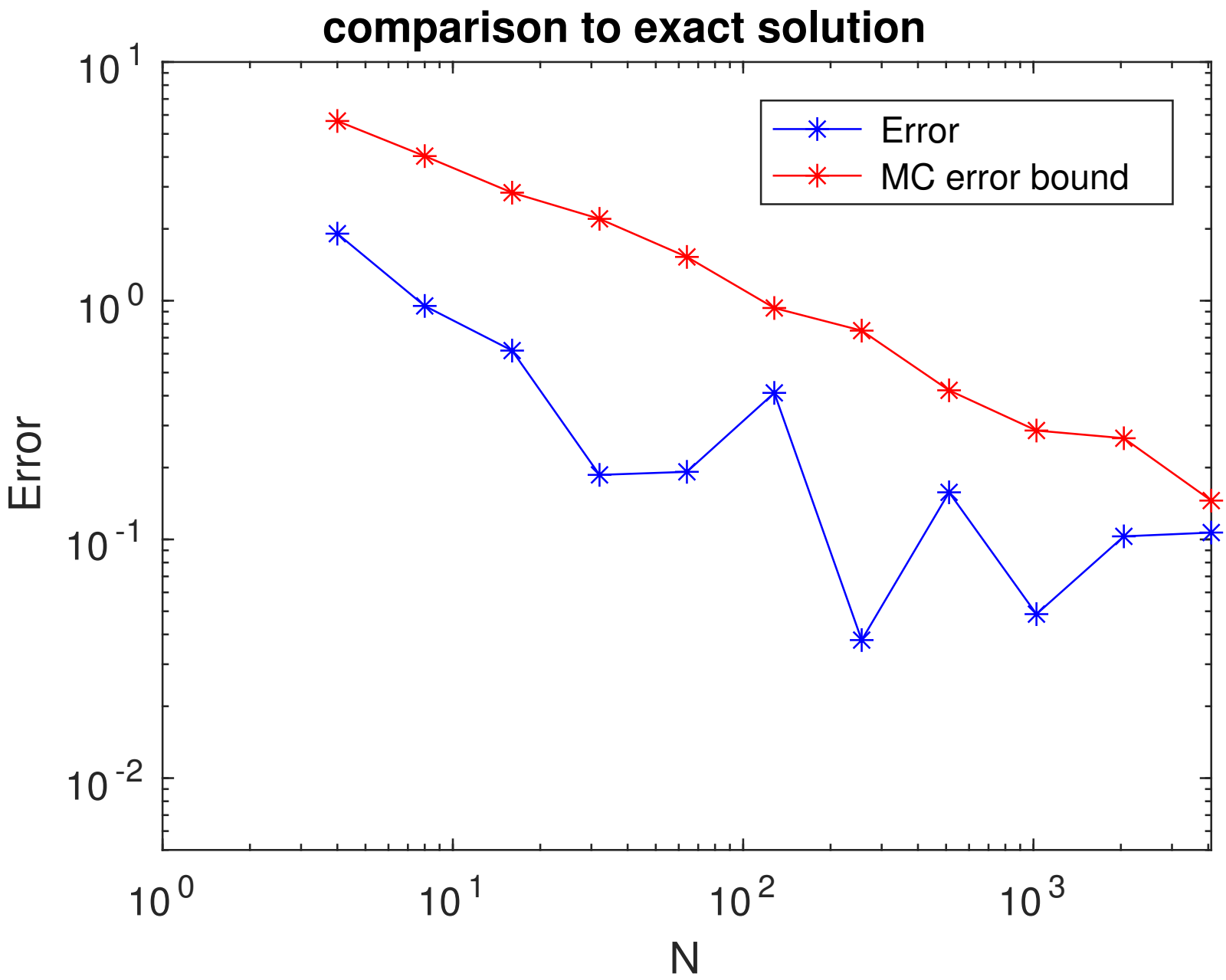
Usual European call test case based on geometric Brownian motion:

- 128 timesteps so weak error is negligible
- comparison between
  - QMC using Brownian Bridge
  - QMC without Brownian Bridge
  - standard MC
- QMC calculations use Sobol generator
- all calculations use 64 “sets” of points – for QMC calcs, each has a different random offset
- plots show error and 3 s.d. error bound

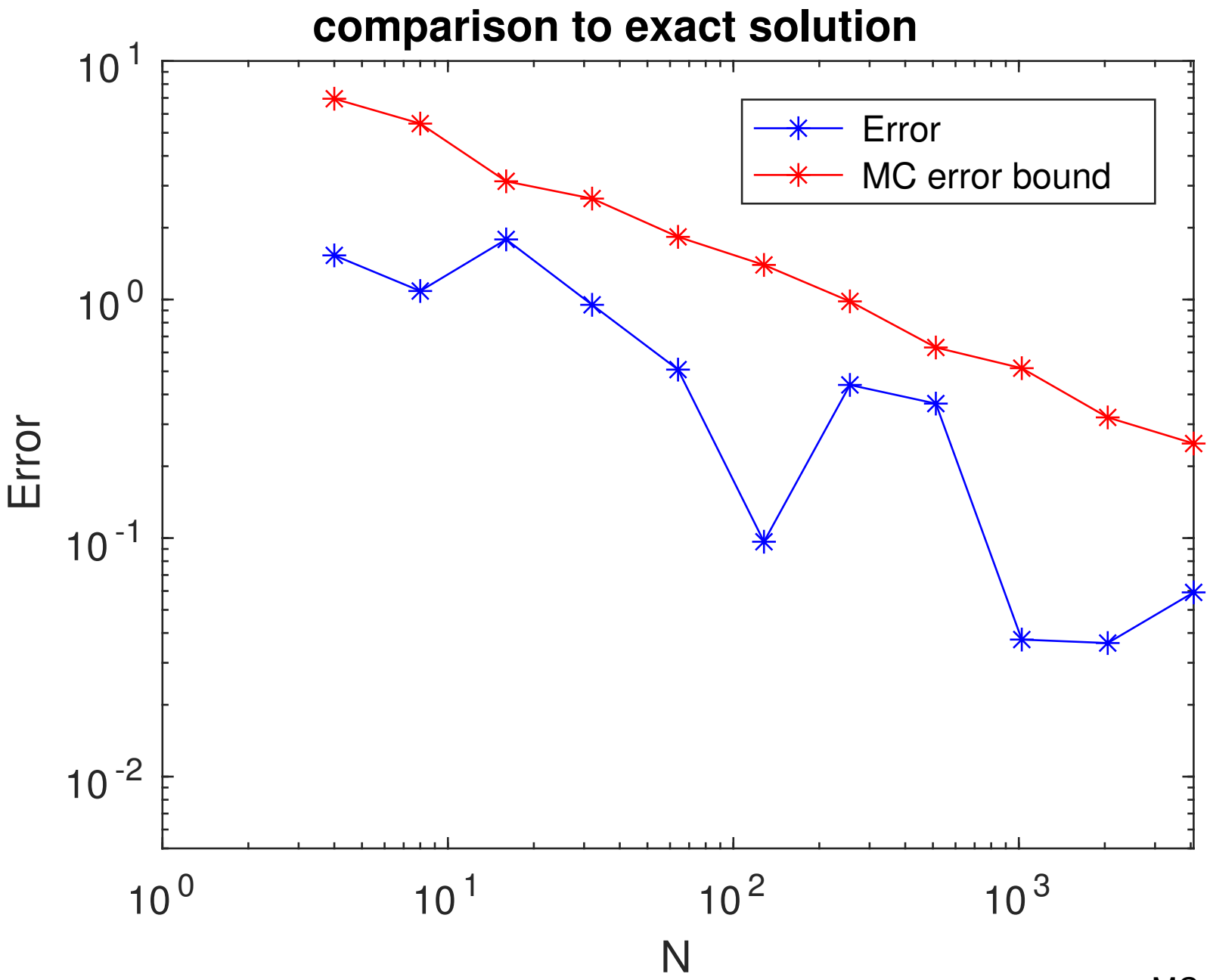
QMC with Brownian Bridge



QMC without Brownian Bridge



Standard Monte Carlo



Final words

- QMC offers large computational savings over the standard Monte Carlo approach
- again advisable to use randomised QMC to regain confidence intervals, at the cost of slightly poorer accuracy
- very important to use PCA or Brownian Bridge construction to create discrete Brownian increments – much better than “standard” approach which is equivalent to Cholesky factorisation of covariance matrix